"Good" and "Bad" Problems and Algorithms

This course devotes significant effort to addressing the question of what constitutes a "good" problem in linear algebra, as opposed to what constitutes a "bad" one, and similarly for the algorithms that we might use to solve those problems. Unfortunately, these are somewhat "fuzzy" questions, and we will certainly not be able to either cleanly or fully answer them here. Moreover, this discussion can be even further clouded by the fact "goodness" or "badness" may vary with what specific *aspect* of a given problem or algorithm we are discussing. For example, some algorithms are very may "good" from the viewpoint of numerical accuracy, but rather "bad" in terms of efficiency.

At this point in the discussion, we shall address this question only from the viewpoint of numerical accuracy, and we shall consider:

> A **problem** "good" when small changes in the "data" of that problem do not significantly change the solution. Conversely, small changes in the data of a "bad" problem may significantly alter the solution.
>
> An **algorithm** "good" when that algorithm is virtually certain to produce accurate solutions for all "good" problems. Conversely, a numerically "bad" algorithm carries significant risks of producing highly inaccurate solutions in the case of at least some good problems.

(Note we should properly call the above statements concepts, since they're really not precise enough to merit the term definition.)

The major insight that we hope you will take away from this section is that while a "bad" algorithm can lead to highly inaccurate solutions to what should have been a simple problem, there is likely **no** algorithm that will adequately solve a *problem* that is simply bad to begin with.

We shall illustrate these concepts further with some examples. First, consider the system of linear equations:

$$\begin{array}{rcrcl} .0001x_1 & + & x_2 & = & 1 \\ x_1 & + & x_2 & = & 2 \end{array} \tag{1}$$

The solution of this system is geometrically identical to the point of intersection of the two lines shown in Figure 1. Moreover, it seems graphically obvious that slightly moving or rotating either (or both) of the lines will not substantially move the point of intersection. Therefore, we claim that, almost intuitively, this should be a "good" system!

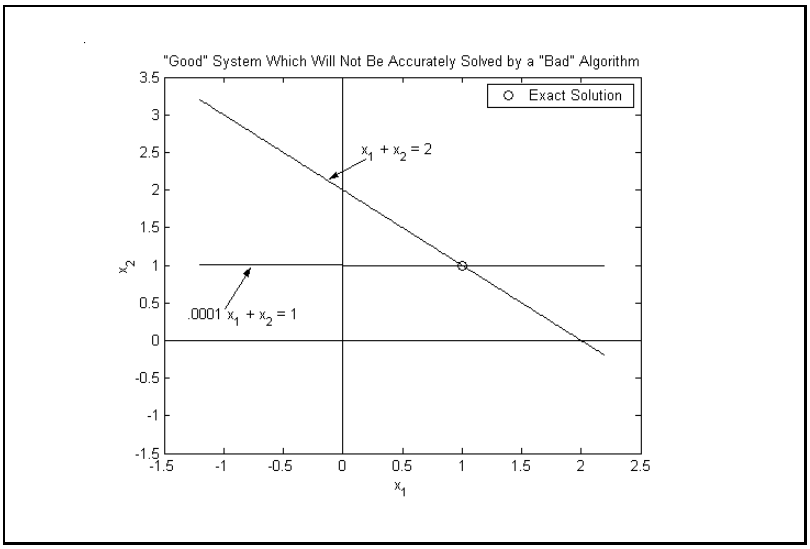

Figure 1

Moreover, numerical experimentation tends to confirm this conclusion, i.e. randomly changing the coefficients in this system by "small" amounts can also be shown not to significantly move the solution. For example, Figure 2 shows how little the true solution moves if we change the system to:

$$\begin{array}{rcrcr} 0.0001\ x_1 & + & 1.0075\ x_2 & = & 1.0292 \\ 1.0087\ x_1 & + & 0.9932\ x_2 & = & 1.9476 \end{array} \tag{2}$$

(Note that all of the coefficeints in (2) vary from those in (1) by less that one percent.)

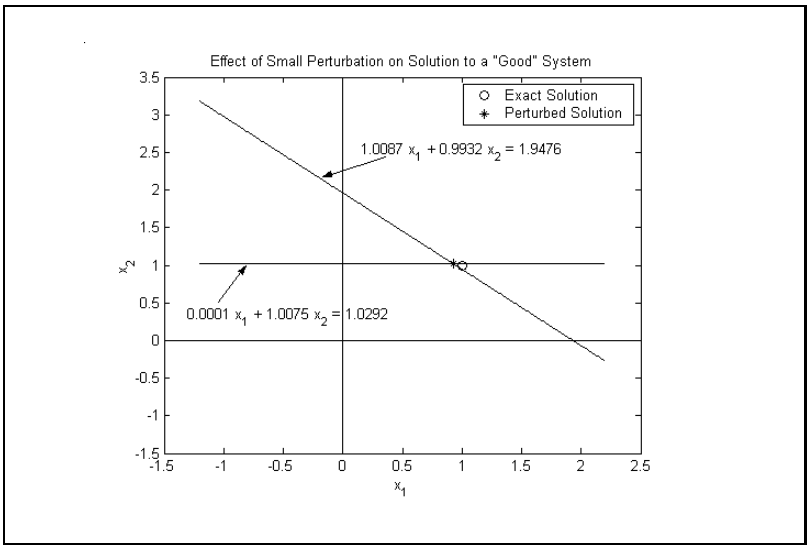

Figure 2

Nevertheless, while we can conclude that (1) should be a good problem, the geometry also plainly indicates that not all algorithms may produce an accurate solution to it. Specifically, we could theoretically use either of the two equations in (1) to solve for one variable in terms of the other. However, the practical truth is very different. Specifically, we can easily show that using the second equation:

$$x_1 + x_2 = 2$$

to solve for either unknown in terms of the other should produce no real problem. This is because, for example, if we rewrite that equation as

$$x_1 = 2 - x_2$$

then $\dfrac{dx_1}{dx_2} = -1$ Therefore, according to the concept of differentials from the elementary calculus, small changes in the numerical value of x_2 will be produce correspondingly small changes in the value of x_1 as well. A similar result occurs if, on the other hand, we rewrite that second equation as:

$$x_2 = 2 - x_1$$

On the other hand, if we rewrite the first equation as

$$x_2 = 1 - .0001x_1$$

then $\dfrac{dx_2}{dx_1} = -.0001$ and so small changes in the numerical value of x_1 will produce miniscule corresponding changes in x_2. However, if were instead rewrite the first equation in (1) as

$$x_1 = 10^4 - 10^4 x_2$$

then $\dfrac{dx_1}{dx_2} = -10^4$**!** Phrased slightly differently, for the second equation, small changes in the numerical value of x_2 result in very large changes in x_1 (an observation that should, in hindsight, be geometrically obvious in Figure 1). Therefore, it would seem clear that **any** algorithm that is equivalent to using the first equation to solve for x_1 in terms of x_2 will very likely be "bad."

Now, contrast the previous example with the system

$$\begin{array}{rcrcl} 1.01\ x_1 & & +0.99\ x_2 & = & 2 \\ 0.99\ x_1 & & +1.01\ x_2 & = & 2 \end{array} \tag{3}$$

The solution to this system is equivalent to the point of intersection of the lines shown in Figure 3. However, as this figure fairly clearly displays, these two lines are almost parallel, and therefore, also fairly clearly from a geometric perspective, any small change in either the slope or intercept of either could significantly move the intersection.

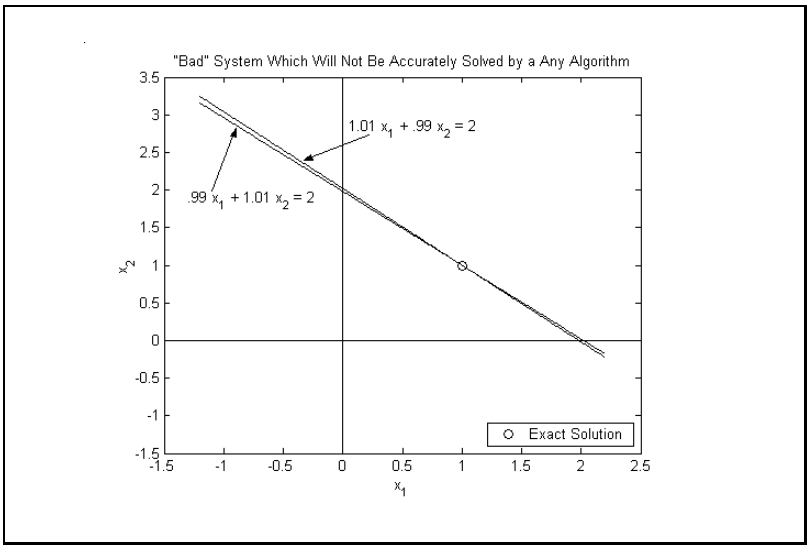

Figure 3

The effect of small changes in the data for this system can also be tested numerically. Figure 4 shows how the solution moves when we replace the original system with:

$$\begin{array}{rcrcl} 1.0036\ x_1 & + & 0.9947\ x_2 & = & 1.9946 \\ 0.9978\ x_1 & + & 1.0137\ x_2 & = & 1.9906 \end{array} \tag{4}$$

Note this is a sizable shift, even though (4) reflects, relatively, no larger change to (3) than we made in going from (1) to (2)). This would seem to confirm our intuitive feeling that (3) is an inherently "bad" problem.

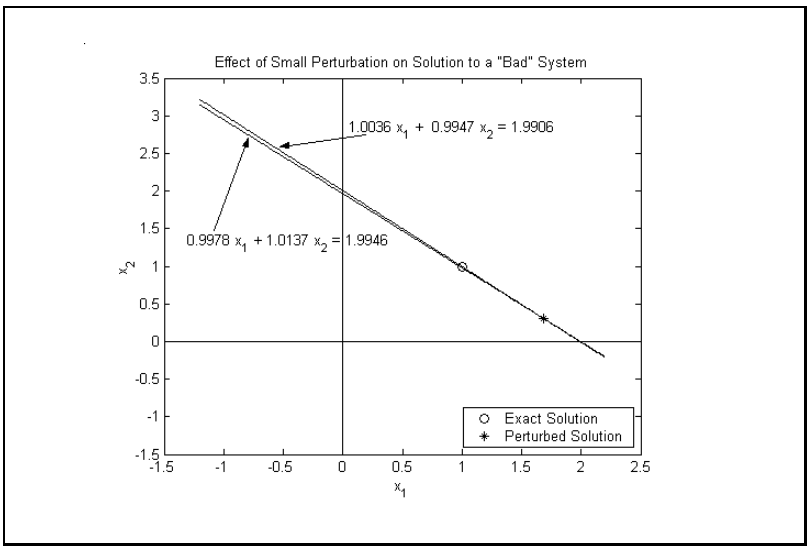

Figure 4